



Release Note ZFRN-012

Z-chipset BIOS version B

ZF Embedded, Incorporated ■ 1052 Elwell Court, Palo Alto, CA 94303 ■ Tel: 650 965-3800 ■ Fax: 650 965-4050

Summary

This application note describes the features of the B version of the ZF Z-chipset Pentium® class BIOS. ZF will continue to develop and enhance the BIOS, and will provide updates to all customers as the newer versions are released.

Information contained in this document:

- DOS management
- Internal Resident Flash Management
- BIOS: Basic Functionality, Features and Future Release Enhancements
- Custom Configuration Screen
- Release Media: Disk Images/Files.
- Flash Architecture
- BIOS Installation process
- Windows 95/98 Installation process
- Rain utility
- ELO Touch installation for Win 95/98
- Post Code Listing

DOS Management

The size of the flash disk currently allows DOS to be loaded on the system as part of a SYS process common to all diskette or hard drives. After formatting the drive, the command **SYS** "flash drive letter" will transfer the system files from the diskette to the RFD. As an example, if the source of the operating system is a diskette in the A drive and you have defined the internal flash as the B drive in the CMOS setup screen, the command **SYS B:** typed at the A: prompt will transfer the files to the RFD.

Internal Resident Flash Management

The internal resident flash does not operate under Win 95/98. The protected mode driver has not been developed. In order to run or install Win 95/98 disable the FLASH in the BIOS Setup screen. This insures that the Virtual System Architecture allows SW restarts using Port 92, Port 64 or a far jump to FFFF:0. Thus, in order to insure that the system is always in the right state you must disable the internal RFD if you plan on using a SW RESTART as part of your design.

BIOS

Basic functionality

The chipset contains embedded video support as well as numerous HW devices such as additional serial ports and USB. These features must be supported in both BIOS and the operating system that you plan using on the board. For instance, although USB HW is present on the board the WIN 95 OS does not support this feature. One major difference from the previous BIOS releases performed by ZF is that this BIOS image is 256K and it supports TFT flat panels with resolutions of 800x600 with no additional video BIOS. As with our previous releases we offer a full complement of drivers, tools and instructions to manage the BIOS installation and HW initialization on our boards. The ZFHOST and ZFUTILS tools have now been extended to support all of our platforms from the 386 to this latest Pentium® class device. We are constantly striving to improve our products and will continue to release additional enhancements.

Features

The version B release supports the following features:

- The entire flash can be updated remotely using the ZFHOST utility, including system BIOS, CMOS setup, and Resident Flash Disk (RFD). There is also a built in capability to copy the information in the flash to your host computer, thereby generating images for manufacturing and field support use. The system BIOS, CMOS setup, and RFD are updated separately.
- ZFUTIL utility performs the same tasks as those present in the ZFHOST utility using local resources vs. the host computer resources used with the ZFHOST program except the RFD update. The system BIOS, CMOS setup, and RFD can be updated from a file, uploaded to a file, or erased using the line command of the ZFUTIL utility.
- Automatic detection of ZF board configurations allows the manipulation of the external devices on the boards with the CMOS CUSTOM configuration screen.
- CMOS setup custom options screen from where additional options can be configured.

Future Release Enhancements

1. The Custom Configuration screen will be expanded to support field programmable registers that will manage:
 - Various Byte-wide socket configurations
 - Additional RS232 serial ports interrupt assignments
 - RS485 reassignment and mode control
 - Watchdog timer enable
2. Interrupt 15 Watchdog timer support
3. Interrupt 15 APM support. Currently available is a utility that minimizes power consumption called RAIN. Please refer to information contained in RAIN section of this document.
4. Windows 98 support. Currently the Windows 98 BIOS support does not correctly manage the following:
 - Internal RFD.
 - Advanced power management

5. Linux support. Many of our customers are turning to Linux as their operating system of choice. The BIOS fully supports Linux, however, the OS does not have the following HW drivers:
 - Internal RFD.
 - Audio support.
 - Universal Serial Bus (USB)
6. The BIOS currently does not support UDMA hard drives.
7. The Virtual Architecture present in the chipset transforms many of the features formerly only available in HW into firmware. One side effect is that video comes up slightly latter in the boot process. This delay coupled with the much higher processing speed makes accessing the CMOS settings via the delete key somewhat difficult. The correct timing to press the DEL key is to press the key ONCE approximately one second after the NUM LOCK/CAPS LOCK/SCROLL LOCK LED's light up. Future releases of the code will improve the CMOS access mechanism.
8. User BIOS extensions resident in flash are not supported.

Custom Configuration Screen

This screen allows you to customize the BIOS functionality to your particular application. It eliminates the need to have various flavors of BIOS for the different hardware settings you may need. Some of the items that can be customized are the following (the underlined option is the default):

- Console redirection – Probe COM1-4, Always COM1-4, Disabled.
- Redirection Speed – baud rate for redirection (recommend 19.2K baud)
- System Display - Allows selection of type of display. Default CRT.
- Infrared @ COM1 – Disabled.

Release Media

The distribution media containing the Rev B BIOS release consists of 3.5" floppy diskettes. This media contains the following:

ZFUTIL ver. 1.92

ZFUTIL.EXE is a utility for the all ZF board and module products. It was designed to be an interactive DOS-environment tool. It supports two modes of operation: a) command line mode for use during re-direction or run from a batch program for automatic upgrades; b) an interactive mode to use if a keyboard and video device are present in the system. It contains on-line help, accessed at the bottom of the ZFUTIL screen giving the user hints, explanation of menu options, etc. This utility contains the functionality of individual *OEMmodule 386* utilities such as FLSHBIOS, RFD-FILE etc. Type *ZFUTIL* for parameter usage details. When you want to load the system BIOS, the utility defaults to ZF.ABS, you can either rename the appropriate file to use this name or type in the full path and filename at the update prompt.

ZFHOST ver. 4.6

ZFHOST.EXE is a utility for all ZF board and module products. It was designed to be an interactive DOS-environment tool. In this context HOST is a standalone PC that can be used as a tool that provides a serial

connection to the TARGET (device containing the OEMmodule) system. It supports a menu driven mode that uses all the resources of the host PC (video, keyboard, floppy/IDE drives, etc) to perform the tasks shown below. It contains on-line help, accessed at the bottom of the ZFHOST screen giving the user hints, explanation of menu options, etc. This utility contains the functionality previously accomplished using the DOWNLOAD procedure on the 386 module.

The functions present in this utility are the following:

- Get Target Attention
- Target System Information
- Test Target Memory
- Continue loading OS on Target
- Update System BIOS
- Update Video BIOS (not applicable to Z-chipset)
- Upload Flash from file (*)
- Download Flash from file (*)
- Erase Flash Block (*)
- Read Target CMOS to File
- Program Target CMOS from File
- Download RFD Image from Target
- Upload RFD Image to Target
- Show Max message size
- Exercise link
- Show Target Drives (*)

(*) not available on this release

Note: These versions of ZFUTIL and ZFHOST have been upgraded to work with rev. B of the Z-chipset BIOS.

Setup Files

Tables 1 and 2 identify the files you will find on the Rev B release diskettes:

File Name	Description
ZFRN-012.PDF	Release notes. Read this file for important last-minute information.
ZFUTIL.EXE	Performs the functionality found before in the various ZF utilities mentioned below: a) INSTALL.BAT. Batch file. Reformats internal Flash disk, loads new BIOS, and default CMOS configuration. b) UPDATE.BAT. Batch file. Loads the new BIOS, and default CMOS configuration. c) CMOSFILE.EXE. Utility that creates a binary file containing the current CMOS configuration, CMOSFILE.BIN . This file can be used to update the CMOS from a floppy drive. d) FILECMOS.EXE. Utility to copy CMOSFILE.BIN to CMOS memory.
ZFHOST.EXE	A menu driven utility that can be used to update the module flash from a remote computer. The capabilities include downloading the RFD, CMOS, video BIOS, and system BIOS. The extraction of data from the flash can also be performed with this utility.
Cyrix Drivers	National/Cyrix Corporation Win 98 drivers, Realtek network drivers and DOS CD-ROM drivers. (on separate diskettes)
Rain Utility	Rain power management utility.
DR_DOS.EXE	Self-extracting ZIP file with a RFD image with Caldera DR-DOS SYS'ed RFD. Use this RFD image when downloading using ZFHOST.EXE.
CRT_CMOS.586	CMOS setup image with the Flash Disk as Drive A: and no Drive B: or Drive C:.. System display is set as CRT only.
LCD_CMOS.586	CMOS setup image with the Flash Disk as Drive A: and no Drive B: or Drive C:.. System display is set as 800 x 600 TFT flat panel.
ZCHIP.ABS	Default flavor of the BIOS image.

Table 1. DR DOS Files Diskette

Z-chipset Version B BIOS Release Notes

File Name	Description
ZFRN-012.PDF	Release notes. Read this file for important last-minute information.
ZFUTIL.EXE	Performs the functionality found before in the various ZF utilities mentioned below: a) INSTALL.BAT. Batch file. Reformats internal Flash disk, loads new BIOS, and default CMOS configuration. b) UPDATE.BAT. Batch file. Loads the new BIOS, and default CMOS configuration. c) CMOSFILE.EXE. Utility that creates a binary file containing the current CMOS configuration, CMOSFILE.BIN . This file can be used to update the CMOS from a floppy drive. d) FILECMOS.EXE. Utility to copy CMOSFILE.BIN to CMOS memory.
ZFHOST.EXE	A menu driven utility that can be used to update the module flash from a remote computer. The capabilities include downloading the RFD, CMOS, video BIOS, and system BIOS. The extraction of data from the flash can also be performed with this utility.
Cyrix Drivers	National/Cyrix Corporation Win 98 drivers, Realtek network drivers and DOS CD-ROM drivers. (on separate diskettes)
Rain Utility	Rain power management utility.
GS_DOS.EXE	Self-extracting ZIP file with a RFD image with General Software DOS SYS'ed RFD. Use this RFD image when downloading using ZFHOST.EXE.
CRT_CMOS.586	CMOS setup image with the Flash Disk as Drive A: and no Drive B: or Drive C:. System display is set as CRT only.
LCD_CMOS.586	CMOS setup image with the Flash Disk as Drive A: and no Drive B: or Drive C:. System display is set as 800 x 600 TFT flat panel.
ZCHIP.ABS	Default flavor of the BIOS image.

Table 2. GSI DOS Files Diskette

Flash Architecture

The board has a total of 2Mbytes of internal flash memory that is organized in 32 individual 64Kbyte pages. Flash utilization and starting addresses are documented in table 3.

Starting Address	Pages used	Function
80000000h	.5	Reserved
80008000h	.5	Reserved (for Extension ROM for C8000h-CFFFFh)
80010000h	1	Reserved for user extension ROM's (D0000h-DFFFFh)
80020000h	.5	Reserved for factory use
80028000h	.5	CMOS Flash backup (512bytes used)
80030000h	1	Reserved for external debugger
80040000h	24	RFD
801C0000h	4	System BIOS, VSA and Video

Table 3. Flash Utilization

BIOS Installation process

The new BIOS can be installed on your target system using two separate methods. The exact method you choose will be dependent upon the resources you have available in your system.

Method A

If your system has a floppy, IDE or Compact flash device, simply copy the files in this release to the drive you have available. Run the ZFUTIL program with the `-i` option, if you have a keyboard and monitor attached to your system. Use the arrow keys to navigate to the BIOS menu option and press Enter to select the process to be performed from the menu. Run the ZFUTIL program in the line mode command, if you are in re-direction mode. When you want to load the system BIOS, the utility defaults to ZF.ABS. You can either rename the appropriate file to use this name or type in the full path and filename at the update prompt.

Method B

If your system does not have a floppy, IDE or Compact Flash (or equivalent) device, you can use the serial port to download the code. Simply install the download cable on COM1 (or the COM port specified in your custom configuration screen) and run the ZFHOST program on the host computer using the cursor keys to select the MENU item to upgrade the system. Specifications for a Download cable are provided in the System Card 5MX Data Book.

Windows 98 Installation process

Before starting make sure you have the "Win98 CD" and the ZF floppy with Cyrix Media GX Certified Win9x Drivers 4.0.exe and Realtek 8139A Ethernet controller driver files for Win98 that are part of this release. It is also preferable to start at this time with an unformatted drive to insure there are no legacy compatibility issues with drives formatted using different HW.

Power up the SCX-3 motherboard and enter the BIOS Setup using the "DEL" key. Insure that no drive is assigned to the Flash Disk. Select drive C: as IDE 0, PM. Set the BOOT ORDER to: Boot 1st Drive A, Boot 2nd Drive C:. Attach a serial mouse to COM1.

Boot from the "Win98 Startup Disk" and enable CDROM Support.

Note: If the SCX-5MX board hangs while detecting the CDROM device 01 continue as follows: Replace existing OAKCDROM.SYS driver on the startup floppy with the DOS CDROM driver in present in the release diskettes. It also is available on the Internet from different web sites of CDROM makers (such as Mitsumi or Samsung). Rename the new driver OAKCDROM.SYS. A third option is to use the Win98 Installation service pack available from Microsoft. Reference: "How to Use Real-Mode CD-ROM Drives from Windows 98 Startup Disk" Internet: <http://support.microsoft.com/support>. After replacing the CDROM driver, reboot the startup disk.

Type **<drive>**: and press **ENTER**. Use the new CDROM drive letter, normally E.

Type **setup** and press **ENTER**, press **ENTER** once more to continue.

Note: If your hard disk is unformatted it will be formatted now. Microsoft ScanDisk will check the hard disk. If ScanDisk ran, press **<X>** to start installation. In the beginning your mouse will not work. Use **TAB** key and **ENTER** to jump between screen buttons.

Follow the Windows Setup instructions. You will be asked to reboot the computer a few times. If the system doesn't start after reboot, recycle the power.

Cyrix Drivers Installation

After the Windows Setup, run "Cyrix Media GX Certified Win9x Drivers 4.0" exe file from the floppy drive. Press "yes" to agree the License Agreement and follow installation instructions. Select "Typical Installation", default folder will be C:\Program Files\Cyrix Corporation. Click "Finish" and reboot the system. Once the system has booted execute the following steps:

DC(Double Click) "My Computer" > DC "Control Panel" > DC "Add New Hardware" > click "Next" > click "Next" > *Is the device that you want install listed below* Select "No, the device isn't in the list" > Click "Next" > Select "No, I want select the hardware from a list" > click "Next" > *Select the type of hardware you want to install* Select "Sound, Video and Game Adapters" > click "Next" > Select "Legacy Audio" > click "Next" > Windows reports that the driver file cannot be found, browse the folder with the Cyrix drivers you previously installed from floppy (default C:\Program Files\Cyrix Corporation) > Select file "GXAUDIO.inf" > click "OK"

Realtek 10/100 Ethernet controller installation

To activate the on-board Ethernet controller execute the following sequence:

Double click (DC) "My Computer" > DC "Control Panel" > DC "Add new hardware" > click "Next" > click "Next", *Windows will search for PnP devices*, 4 devices will be found: Advanced Power Management Support, PCI Ethernet Controller, Standard Dual PCI IDE Controller, Standard PS/2 Port Mouse > Select "PCI Ethernet Controller" > click "Next" > click "Finish" > Select "PCI Ethernet Controller Properties" > select "Reinstall Driver" > select "Search for a better driver" > click "Next" > Browse location of Realtek Win98 drivers, default A:\Realtek\Win98 > Windows will show *Updated drivers*, select "Realtek RTL8139(A/B/C/8130) PCI Fast Ethernet" > click "Next" > click "Next" > browse back to the same location and location of Win98 setup files if requested, default location E:\win98

Note: Sometimes you will loose the floppy disk controller during Windows PnP search. Then you have to go back to *Windows will search for PnP devices* and reinstall Floppy disk and copy the Realtek driver files to the C: disk and install them from C:.

Reboot after Cyrix and Realtek hardware installation. Several hardware components will be found. SCX-5MX is ready to use.

Rain Utility

The Rain utility is a program that will put the processor in a HALT state whenever it detects that the processor is idle. This feature will save power and also keep the processor cool. It will result in a significant extension of the operating temperature range. Please refer to the README file for additional details.

Installation process. Simply extract the zip file to a temporary directory and run the install.exe file.

The installation automatically detects and optimizes your CPU.

Note: If you run the RAIN.EXE file directly, RAIN won't detect your CPU and might not work at all. Please use the shortcuts created by the installation utility in your start menu, or use the manual optimization.

Manual Optimization: If you would like to manually create a shortcut for Rain, just type “rain - Cyrix6x86MX”

ELO Touchscreen installation for Win 95/98

1. Shutdown Windows 95/98 and turn off your computer.
2. Start your computer. Unzip mm95_200.exe to an empty floppy
3. After Windows loads, insert the MonitorMouse for Windows 95 disk into drive A.
4. Click the Start button, and then click Run.
5. Type "a:\setup" in the space provided and press Enter. If you downloaded the driver, Browse for setup.exe in the directory to which you extracted the driver files.
6. Follow the directions on the screen.
7. Choose your Elo touchscreen controller from the list:
"Elo 2310 or 2300 Smart Serial Controller on COM2"
8. Complete the Setup program.
9. Restart Windows when prompted. The touchscreen calibration program will automatically run when Windows starts up. Touch each of the three targets as they appear on the screen. Click Yes when the cursor lines up correctly with your finger.

Uninstalling ELO touchscreen for Windows 95

1. Click the Start button, point to Settings, and then click Control Panel.
2. Double-click the Add/Remove Programs icon.
3. Select MonitorMouse for Windows 95 from the list of uninstallable programs on the Install/Uninstall tab.
4. Click the Add/Remove button then follow the directions on your screen.
5. Restart Windows to complete the uninstall process.

Z-chip Post Code Listing

The following list contains symbol definitions associated with POST progress reporting in chronological, not numerical, order.

POST_STATUS_START = 00h ; POST beginning.

Z-chipset Version B BIOS Release Notes

POST_STATUS_CPUTEST	=	01h	; CPU register test about to start.
POST_STATUS_DELAY	=	02h	; NMIs are disabled; delay starts.
POST_STATUS_DELAYDONE	=	03h	; power-on delay finished.
POST_STATUS_KDBATRDY	=	04h	; kbd BAT done; reading kbd SYS bit.
POST_STATUS_DISABSHADOW	=	05h	; disabling shadowing & cache.
POST_STATUS_CALCKSUM	=	06h	; calcing ROM cksum, wait kbd ctrlr.
POST_STATUS_CKSUMGOOD	=	07h	; cksum okay, kbd ctrlr free.
POST_STATUS_BATVRFY	=	08h	; verifying BAT cmd to kbd ctrlr.
POST_STATUS_KBDCMD	=	09h	; issuing kbd ctrlr cmd byte.
POST_STATUS_KBDDATA	=	0ah	; issuing kbd ctrlr data byte.
POST_STATUS_BLKUNBLK	=	0bh	; issuing pin 23,24 blocking & unblocking.
POST_STATUS_KBDNOP	=	0ch	; issuing kbd ctrlr NOP cmd next.
POST_STATUS_SHUTTEST	=	0dh	; testing CMOS RAM shutdown register.
POST_STATUS_CMOSDIAG	=	0eh	; checking CMOS cksum, updating DIAG byte.
POST_STATUS_CMOSINIT	=	0fh	; initializing CMOS (if req'd every boot).
POST_STATUS_CMOSSTATUS	=	10h	; init CMOS status reg for date/time.
POST_STATUS_DISABDMAINT	=	11h	; disabling DMA, interrupt ctrlrs.
POST_STATUS_DISABPORTB	=	12h	; disabling Port B, disabling video display.
POST_STATUS_BOARD	=	13h	; init board, start auto-mem detect.
POST_STATUS_TESTTIMER	=	14h	; starting timer tests.
POST_STATUS_TESTTIMER2	=	15h	; testing 8254 T2, for spkr, part B.
POST_STATUS_TESTTIMER1	=	16h	; testing 8254 T1, for refresh.
POST_STATUS_TESTTIMER0	=	17h	; testing 8254 T0, for 18.2Hz.
POST_STATUS_MEMREFRESH	=	18h	; starting memory refresh.
POST_STATUS_TESTREFRESH	=	19h	; testing memory refresh.
POST_STATUS_TEST15US	=	1ah	; testing 15usec refresh ON/OFF time.
POST_STATUS_TEST64KB	=	1bh	; testing base 64KB memory.
POST_STATUS_TESTDATA	=	1ch	; testing data lines.
POST_STATUS_TESTADDR	=	20h	; testing address lines.
POST_STATUS_TESTPARITY	=	21h	; testing parity (toggling).
POST_STATUS_TESTMEMRDWR	=	22h	; base 64KB mem read/write test.

; Now we have memory, so we can use a stack to use Pcall, not Rcall.

POST_STATUS_SYSINIT	=	23h	; system init before vector table init.
POST_STATUS_INITVECTORS	=	24h	; init vector table.
POST_STATUS_8042TURBO	=	25h	; reading 8042 for turbo switch setting.
POST_STATUS_POSTTURBO	=	26h	; initializing turbo data.
POST_STATUS_POSTVECTORS	=	27h	; any init after vector table init is next.
POST_STATUS_MONOMODE	=	28h	; setting monochrome mode.
POST_STATUS_COLORMODE	=	29h	; setting color mode.
POST_STATUS_TOGGLEPARITY	=	2ah	; toggle parity before optional video ROM test.
POST_STATUS_INITBEFOREVIDEO	=	2bh	; init before video ROM check.
POST_STATUS_VIDEOROM	=	2ch	; control passed to video ROM.
POST_STATUS_POSTVIDEO	=	2dh	; video ROM returned control.
POST_STATUS_CHECKEGAVGA	=	2eh	; checking for EGA/VGA adapter found.
POST_STATUS_TESTVIDEOMEMORY	=	2fh	; no EGA/VGA found, r/w test of video memory.
POST_STATUS_RETRACE	=	30h	; looking for video retrace signal.
POST_STATUS_ALTDISPLAY	=	31h	; retrace failed, checking alt. display.
POST_STATUS_ALTRETRACE	=	32h	; alt found, checking video retrace signal.
POST_STATUS_VRFYSWADAPTER	=	33h	; compare switches w/actual adapter type.
POST_STATUS_SETDISPMODE	=	34h	; setting display mode.

; Now we have a display. All code that outputs codes at 35h and above
; can use INT 10h to display messages.

POST_STATUS_CHECKSEG40A =	35h	; check ROM BIOS data area at seg 40h.
POST_STATUS_SETCURSOR =	36h	; setting cursor for power-on msg.
POST_STATUS_PWRONDISPLAY=	37h	; displaying power-on message.
POST_STATUS_SAVECURSOR=	38h	; save cursor position.
POST_STATUS_BIOSIDENT =	39h	; display BIOS ident. string.
POST_STATUS_HITDEL =	3ah	; display "Hit to ..." msg.
POST_STATUS_VIRTUAL =	40h	; preparing vm test. vrfy from display memory.
POST_STATUS_DESCR =	41h	; preparing descriptor tables.
POST_STATUS_ENTERVM =	42h	; enter virtual mode for memory test.
POST_STATUS_ENABINT =	43h	; enable ints for diagnostics mode.
POST_STATUS_CHECKWRAP1=	44h	; init data for checking wraparound at 0:0.
POST_STATUS_CHECKWRAP2=	45h	; checking for wrap, find total memory size.
POST_STATUS_HIGHPATTERNS=	46h	; write extended memory test patterns.
POST_STATUS_LOWPATTERNS =	47h	; write conventional memory test patterns.
POST_STATUS_FINDLOWMEM=	48h	; finding low memory size from patterns.
POST_STATUS_FINDHIMEM =	49h	; finding high memory size from patterns.
POST_STATUS_CHECKSEG40B =	4ah	; check ROM BIOS data area again.
POST_STATUS_CHECKDEL =	4bh	; check for , clear low mem for soft reset.
POST_STATUS_CLREXTMEM =	4ch	; clearing ext mem for soft reset.
POST_STATUS_SAVEMEMSIZE =	4dh	; saving memory size.
POST_STATUS_COLD64TEST=	4eh	; on cold boot, display 1st 64KB memtest.
POST_STATUS_COLDLOWTEST =	4fh	; on cold boot, test all of low memory.
POST_STATUS_ADJUSTLOW =	50h	; adjust memsize for 1K usage.
POST_STATUS_COLDHITEST =	51h	; on cold boot, test high memory.
POST_STATUS_REALMODETEST=	52h	; prepare for shutdown to real-mode.
POST_STATUS_ENTERREAL =	53h	; saved regs & memsize, entering real-mode.
POST_STATUS_SHUTDOWN =	54h	; shutdown successful, restoring codepath.
POST_STATUS_DISABA20 =	55h	; disabling A20 line.
POST_STATUS_CHECKSEG40C =	56h	; checking ROM BIOS data area again.
POST_STATUS_CHECKSEG40D =	57h	; checking ROM BIOS data area some more.
POST_STATUS_CLRHITDEL =	58h	; clear the "Hit " message.
POST_STATUS_TESTDMAPAGE =	59h	; test DMA page register.
POST_STATUS_VRFYDISPMEM =	60h	; verify from display memory (???).
POST_STATUS_TESTDMA0BASE=	61h	; test DMA0 base register.
POST_STATUS_TESTDMA1BASE=	62h	; test DMA1 base register.
POST_STATUS_CHECKSEG40E =	63h	; checking ROM BIOS data area again.
POST_STATUS_CHECKSEG40F =	64h	; checking ROM BIOS data area some more.
POST_STATUS_PROGDMA =	65h	; programming DMA ctrllrs 0 & 1.
POST_STATUS_INITINTCTRL =	66h	; initializing INT ctrllrs 0 & 1.
POST_STATUS_STARTKBDTEST=	67h	; starting keyboard test.
POST_STATUS_KBDRESET =	80h	; issuing reset cmd & clrng output buffer.
POST_STATUS_CHECKSTUCKKEYS =	81h	; check for stuck keys & issue test cmd.
POST_STATUS_INITCIRCBUFFER =	82h	; initializing circular buffer.
POST_STATUS_CHECKLOCKEDKEYS=	83h	; check for locked keys.
POST_STATUS_CHECKMEMSIZEMISMATCH =	84h	; check for memsize mismatch (CMOS/BIOSDATA).
POST_STATUS_PASSWORD =	85h	; check for pswd or bypass setup.
POST_STATUS_BEFORESETUP =	86h	; pswd checked. do pgming before setup.
POST_STATUS_CALLSETUP =	87h	; call the setup module.
POST_STATUS_POSTSETUP =	88h	; back from setup, clr screen.
POST_STATUS_DISPPWRON =	89h	; display power-on screen message.
POST_STATUS_DISPWAIT =	8ah	; display "Wait..." message.
POST_STATUS_ENABSHADOW=	8bh	; do system & video BIOS shadowing.
POST_STATUS_STDCMOSSETUP=	8ch	; load standard setup params into BIOSDATA.
POST_STATUS_MOUSE =	8dh	; check and initialize mouse.

Z-chipset Version B BIOS Release Notes

POST_STATUS_FLOPPY = 8eh ; check floppy disks.
POST_STATUS_CONFIGFLOPPY= 8fh ; configure floppy drives.
POST_STATUS_IDE = 90h ; check hard disks.
POST_STATUS_CONFIGIDE = 91h ; configure IDE drives.
POST_STATUS_CHECKSEG40G = 92h ; checking ROM BIOS data area again.
POST_STATUS_CHECKSEG40H = 93h ; checking ROM BIOS data area some more.
POST_STATUS_SETMEMSIZE = 94h ; setting base & ext mem sizes.
POST_STATUS_SIZEADJUST = 95h ; memsize adjusted for 1K, verifying disp mem.

; Initialize any ROM BIOS extensions.

POST_STATUS_INITBEFOREC8000 = 96h ; initialization before calling C800h.
POST_STATUS_CALLC8000 = 97h ; call ROM BIOS extension at C800h.
POST_STATUS_POSTC8000 = 98h ; processing after extension returns.

; Setup serial ports, parallel ports, NPX, keyboard, cache, wait states.

POST_STATUS_TIMERPRNBASE= 99h ; configuring timer data area, printer base addr.
POST_STATUS_SERIALBASE = 9ah ; configuring serial port base addrs.
POST_STATUS_INITBEFORENPX = 9bh ; initialization before coprocessor test.
POST_STATUS_INITNPX = 9ch ; initializing the coprocessor.
POST_STATUS_POSTNPX = 9dh ; processing after coprocessor initialized.
POST_STATUS_CHECKLOCKS= 9eh ; check ext kbd, kbdID, numlock settings.
POST_STATUS_ISSUEKBDID = 9fh ; issue keyboard ID command next.
POST_STATUS_RESETID = 0a0h ; kbd ID flag reset.
POST_STATUS_TESTCACHE = 0a1h ; do cache memory test.
POST_STATUS_DISPSOFTERR = 0a2h ; display any soft errors.
POST_STATUS_TYPERMATIC = 0a3h ; set keyboard typematic rate.
POST_STATUS_MEMWAIT = 0a4h ; program memory wait states.
POST_STATUS_CLRSCR = 0a5h ; clear screen.
POST_STATUS_ENABPTYNMI = 0a6h ; enable parity and NMIs.

; Initialize ROM BASIC if available.

POST_STATUS_INITBEFOREEE000 = 0a7h ; initialization before calling E000h.
POST_STATUS_CALLEE000 = 0a8h ; call ROM BIOS extension at E000h.
POST_STATUS_POSTEE000 = 0a9h ; processing after extension returns.

; Boot operating system.

POST_STATUS_DISPCONFIG = 0b0h ; display system config box.
POST_STATUS_INT19BOOT = 00h ; call INT 19h bootstrap loader.

; Additional paths.

POST_STATUS_LOWMEMEXH= 0b1h ; test low memory exhaustively.
POST_STATUS_EXTMEMEXH = 0b2h ; test extended memory exhaustively.

Release Note Revision History:

Revision A 8/99 (Initial Release)

SMX/386, OEMmodule 386, OEMmodule 486 are trademarks of ZF Embedded, Incorporated.
All other trademarks and registered trademarks are the property of their respective owners.



ZF Embedded, Incorporated

1052 Elwell Court • Palo Alto, California 94303

Tel: (650) 965-3800 Fax: (650) 965-4050

Home Page: <http://www.zfmicro.com>

Email: support@zfmicro.com

ZF Embedded assumes no responsibility or liability for any errors or inaccuracies contained in this document, nor for the use of the information or any portion of this information in any application, including any claim for copyright or patent infringement or direct, indirect, special or consequential damages. There are no warranties extended or granted by this document. The information herein is subject to change without notice from ZF Embedded, Incorporated. ©1998 ZF Embedded, Incorporated. All rights Reserved.